### Module Learning Outcomes

Learning
Analysis
Enquiry
Application

# 1) Porting Your ERD and Normalised Data into QSEE

Using the Employees on a Project (Normalisation Exercise 1)

A company has a number of **projects** on the go at any one time and each project will have several **employees** working on it. The employees do not necessarily stay for the full duration of the project but instead join it when they are required and leave when their particular specialist skills are no longer needed. Each employee is paid a salary and this is determined by the pay grade they are on.

UNF sample data is shown below:

<u>Project</u> Code	Project Description	Employee Number	Employee Firstname	Employee Surname	Pay Grade	Salary	Date Joined Project	Months Allocated To Project
	Allied							-
A21	Carpets	12	Tom	Jones	7	30000	05/05/2000	12
		56	Andrea	Murray	5	24000	05/05/2000	9
		60	Bob	Roberts	6	27000	01/07/2000	10
G02	Game	25	Jenny	Smith	5	24000	04/05/2002	`18
l11	Iceland	12	Tom	Jones	7	30000	31/12/2000	10
S03	Sainsburys	12	Tom	Jones	7	30000	01/01/2000	12
		56	Andrea	Murray	5	24000	31/05/2000	6
Z04	Zavvi	56	Andrea	Murray	5	24000	01/08/2000	12

#### **3NF version of the database**

Project (Proj Code, Proj\_Desc)

**Emp\_On\_Project** (<u>Proj Code</u>, <u>Emp No#</u>, Date\_Joined\_Proj, Months\_Allocated\_To\_Proj)

**Employee** (Emp No#, Emp\_fname, Emp\_sname, Grade#)

Pay\_Structure (Grade#, Salary#)

# Some Data Types in MySQL

What type of data?	Туре	Usage
Date	Date	MySQL displays DATE values
		in 'YYYY-MM-DD' format
True or False	TinyInt	Boolean – true or false <sup>1</sup>
-128 to 127 <sup>2</sup>	TinyInt	Whole number
-32768 to 32767 <sup>3</sup>	SmallInt	Whole number
-2147483648 to 2147483647 <sup>4</sup>	Integer/ Int	Whole number
Numbers with a decimal place – such	Real	Numbers with a decimal place (real
as currency		is the alias for Double Precision)
A fixed number of characters	VarChar(m)	m represents the maximum length in characters

Full breakdown of data types available here: http://dev.mysql.com/doc/refman/5.0/en/data-type-overview.html

Note: To make a field unsigned (see footnotes 2 to 4 below) – you need to amend the SQL generated by QSEE before it is added to MySQL (believe me it is simpler than the alternative).

# Building an ERD from the 3NF Entities and Attributes

- Create a new project in QSEE
  - QSEE SuperLite [Help]

     Image: Project
     Edit
     View
     Window
     Help

     Image: New Project
     Ctrl+ N

     Image: Open Project...
     Ctrl+ O
- Right click on the project title and select "Add an Entity Relationship model"



Name your project "EmployeeProject"

 $<sup>^1</sup>$  Bit is the better option but not included in QSEE – you would have to change it when you had imported the SQL into MySQL

<sup>&</sup>lt;sup>2</sup> 0 to 255 (unsigned)

<sup>&</sup>lt;sup>3</sup> 0 to 65535 (unsigned)

<sup>&</sup>lt;sup>4</sup> 0 to 4294967295 (unsigned)

Name Details	6-CASE	
Name		
EmployeeProject	ct	
	ОК	Cancel

- Expand the project title so you can see your ERD:
  - 🖃 💼 untitled

EmployeeProject

- Right click on the ERD name and select **Options** then change the following:
  - Use SSADM LDM Notation [Check]
  - Verbose Cardinality [Uncheck]
  - Verbose Code Generation [Uncheck]

V	Use SSADM LDM notation
	Verbose Cardinality
	Hide Attributes
	Hide Source/Dest labels
	Hide Relationship labels
	Bold Relationships

Verbose Code Generation

- Right click on the ERD name and select **Options** (again) then click on Foreign Key Generation (global) and change the following:
  - Auto Generated [Uncheck]

Foreign Key Generation
Prefix Name
Auto Generated
Do Not Generate Foreign Keys
OK Cancel

• Right click -> **Target** then select **MySQL 4.1** 

Target		SQL2
Options	►	ORACLE 7
😵 Reverse Engineer		ORACLE 8 MySOL 4.0
View Owning Object		MySQL 4.1
Add Free Text		SQL Server

**Note**: You are about to create the following ERD diagram – it is included in full here so that as you go you can lay it out correctly.



# <u>3NF version of the database (included again so that you can compare it with the QSEE ERD below)</u>

Project (Proj Code, Proj\_Desc)
Emp\_On\_Project (Proj Code, Emp No#, Date\_Joined\_Proj,
Months\_Allocated\_To\_Proj)
Employee (Emp No#, Emp\_fname, Emp\_sname, Grade#)
Pay\_Structure (Grade#, Salary#)

**Note**: You do not need to add fields that are foreign keys. QSEE assumes you want them by simply adding a relationship between two entities. Foreign keys are not shown in QSEE though they do appear when the data is exported to MySQL.

**Note**: The Emp\_On\_Project entity has a dotted outline – this is because it has no primary key assigned at this point (because the primary key in this table is a composite key based on Emp\_No and Proj\_Code – and they are foreign keys from their respective tables).

# Project

Project (Proj Code, Proj\_Desc)

What is the entity name?

For each attribute:

- Is the attribute(s) a primary key?
- What data type does it need to be look at the sample data?
- For fields other than the primary key can the field be blank?
  - **Right click** in the large area below the ERD diagram name (as shown) and select **Add Entity**



You will then see a dotted rectangle which will become your entity
 <u>EmployeeProject</u>



• Left click when you are satisfied with its position on the ERD diagram. You will then be asked to enter the name of the entity. Call it "Project"

Name Details	
Name	
Project	
	OK Cancel

Right click on Project then select Add Attribute



• Enter the following attribute details as highlighted below:

Attribute Details			
Name Proj_Code			
Type VARCHAR			•
Size 3			
Key	💌 Not Null	🗹 Uniqu	16
		ОК	Cancel

• Right click on Project again and select Add Attribute

Attribute Details	
Name Proj_Desc	
Type VARCHAR	[
Size 30	
☐ Key	
OK Cancel	

Your diagram should now look like this (showing the two attributes including primary key):

EmployeeProject



### **Pay Structure**

### Pay\_Structure (Grade#, Salary#)

What is the entity name?

For each attribute:

- Is the attribute(s) a primary key?
- What data type does it need to be look at the sample data?
- For fields other than the primary key can the field be blank?
  - **Right click** in the large area below the ERD diagram name (as shown) and select **Add Entity**
  - You will then see a dotted rectangle which will become your entity. Left click when you are satisfied with its position on the ERD diagram.
  - You will then be asked to enter the name of the entity. Call it "Pay\_Structure"
  - Right click on Pay\_Structure then select Add Attribute
    - Name: Grade
    - Type: Tiny Int
    - Key [Check]
  - Right click on Pay\_Structure then select Add Attribute
    - Name: Salary
    - Type: Real
    - Not Null [Check]

# Employee

**Employee** (Emp No#, Emp\_fname, Emp\_sname, Grade#)

What is the entity name?

For each attribute:

- Is the attribute(s) a primary key?
- What data type does it need to be *look at the sample data*?
- For fields other than the primary key can the field be blank?
  - Right click in the large area below the ERD diagram name and select Add Entity
  - You will then see a dotted rectangle which will become your entity. Left click when you are satisfied with its position on the ERD diagram.
  - You will then be asked to enter the name of the entity. Call it "Employee"
  - Right click on Employee then select **Add Attribute** 
    - Name: Emp\_No
    - Type: Integer (ideally SmallInt)
    - Key [Check]
  - Right click on Employee then select Add Attribute
    - Name: **Emp\_fname**
    - Type: VarChar -> Size 30
    - Not Null [Check]
    - Right click on Employee then select Add Attribute
      - Name: Emp\_sname
      - Type: VarChar -> Size 30
      - Not Null [Check]
  - **DO NOT ADD a Grade attribute** instead do the following (which adds the Grade field to the Employee entity though you won't see it in QSEE but it show up in MySQL):
    - Right Click on Pay\_Structure -> Add Relationship
       -> Click on Employee
       Relationship "has"
       Cardinality 1:m

This part of the ERD diagram should now look like this:



### **Employee on Project**

**Emp\_On\_Project** (<u>Proj Code</u>, <u>Emp No#</u>, Date\_Joined\_Proj, Months\_Allocated\_To\_Proj)

What is the entity name?

For each attribute:

- Is the attribute(s) a primary key?

- What data type does it need to be - look at the sample data?

- For fields other than the primary key can the field be blank?
  - Right click in the large area below the ERD diagram name and select Add Entity
  - You will then see a dotted rectangle which will become your entity. Left click when you are satisfied with its position on the ERD diagram.
  - You will then be asked to enter the name of the entity. Call it "Emp\_On\_Project"
  - **DO NOT ADD the Proj\_Code attribute** instead do the following (which adds the Proj\_Code field to Emp\_On\_Project but it show up in MySQL):
    - Right Click on Emp\_On\_Project -> Add Relationship
       -> Click on Project
       Cardinality m:1
  - **DO NOT ADD the Emp\_No attribute** instead do the following (which adds the Emp\_No field to Emp\_On\_Project but it show up in MySQL):
    - Right Click on Emp\_On\_Project -> Add Relationship
       -> Click on Employee
      - Cardinality m:1
  - Right click on Emp\_On\_Project then select Add Attribute
    - Name: Date\_Joined\_Proj
    - Type: Date
    - Not Null [Check]
  - Right click on Emp\_On\_Project then select **Add Attribute** 
    - Name: Months\_Allocated\_To\_Proj
    - Type: TinyInt



### To ensure that the Emp On Project has a composite primary key

Remember Emp\_No and Proj\_Code do not appear in Emp\_On\_Project as they are foreign keys (though they do exist). To make them a composite/joint primary key we need to do the following:

Right click on the relationship between Employee and Emp\_On\_Project and select **Edit Relationship Details**.



Then ensure that the Identifier "Key for related entities" check box is selected.

Relationship Details	
Relationship Name	
Source Name	
Destination Name	
Cardinality C 1:1 C 1:m	
Optional	
At Source	At Destination
Identifier	>
	OK Cancel

Do the same thing for the relationship between Emp\_On\_Project and Proj\_Code.

# 2) Exporting your QSEE model as SQL

In the left hand window right click on EmployeeProject in the left hand window then select "Generate SQL (DDL)"



You will then be presented with the following dialog box.

💐 Save As	×
Save in: 🚺 Day 06 🗨	+ 🗈 📸 📰 🕶
Name	Date modified Ty
EmployeeProject.sql	13/10/2011 09:59 SC
<	4
File name: EmployeeProject.sql	<u>S</u> ave
Save as type: SQL Files (*.sql)	Cancel

**Note**: The file created is not perfect – i.e. it will not load as it is in MySQL! You need to open it in notepad then edit the file and replace all "---" with "-- " and replace all "TYPE=INNODB;" with ";"

#### DO NOT INCLUDE THE QUOTATION MARKS WHEN REPLACING THE TEXT

I am including the ERD again here so you can compare it with the SQL statements below:



# Project SQL

-- Create a Database table to represent the "Project" entity.

### CREATE TABLE **Project**(

Proj\_Code VARCHAR(3) NOT NULL,

### Proj\_Desc VARCHAR(30) NOT NULL,

-- Specify the PRIMARY KEY constraint for table "Project".

-- This indicates which attribute(s) uniquely identify each row of data.

### CONSTRAINT pk\_Project **PRIMARY KEY (Proj\_Code)**

);

# Pay\_Structure SQL

-- Create a Database table to represent the "Pay\_Structure" entity.

### CREATE TABLE **Pay\_Structure**(

Grade	TINYINT	NOT NULL,
		,

### Salary REAL NOT NULL,

- -- Specify the PRIMARY KEY constraint for table "Pay\_Structure".
- -- This indicates which attribute(s) uniquely identify each row of data.

### CONSTRAINT pk\_Pay\_Structure **PRIMARY KEY (Grade)**

);

# Employee SQL

-- Create a Database table to represent the "Employee" entity.

### CREATE TABLE **Employee**(

Emp_No	INTEGER NOT NULL,
Emp_FName	VARCHAR(30) NOT NULL,
Emp_SName	VARCHAR(30) NOT NULL,
Grade	TINYINT NOT NULL,

-- Specify the PRIMARY KEY constraint for table "Employee".

-- This indicates which attribute(s) uniquely identify each row of data.

CONSTRAINT pk\_Employee **PRIMARY KEY (Emp\_No)** 

#### );

ALTER TABLE Employee ADD INDEX (Grade), ADD CONSTRAINT fk1\_Employee\_to\_Pay\_Structure **FOREIGN KEY(Grade) REFERENCES Pay\_Structure(Grade)** ON DELETE RESTRICT ON UPDATE RESTRICT;

### Emp\_On\_Project SQL (where is the key?)

-- Create a Database table to represent the "Emp\_On\_Project" entity.

### CREATE TABLE Emp\_On\_Project(

DATE NOT NULL,
ject TINYINT NOT NULL,
VARCHAR(3) NOT NULL,
INTEGER NOT NULL
Dn_Project PRIMARY KEY (Proj_Code,Emp_No)

#### );

ALTER TABLE **Emp\_On\_Project** ADD INDEX (Proj\_Code), ADD CONSTRAINT fk1\_Emp\_On\_Project\_to\_Project **FOREIGN KEY(Proj\_Code) REFERENCES Project(Proj\_Code)** ON DELETE RESTRICT ON UPDATE RESTRICT;

ALTER TABLE **Emp\_On\_Project** ADD INDEX (Emp\_No), ADD CONSTRAINT fk2\_Emp\_On\_Project\_to\_Employee **FOREIGN KEY(Emp\_No) REFERENCES Employee(Emp\_No)** ON DELETE RESTRICT ON UPDATE RESTRICT;

# 3) Importing your SQL into XAMPP

Launch XAMPP – then click the start buttons next to both the Apache and MySql modules

XAMPP Control Panel v3.1.0 3.1.0 [Compiled: September 20th 2012]										
XAMPP Control Panel v3.1.0 3.1.0										
Service	Module	PID(s)	Port(s)	Actions						
	Apache	3564 6244	80, 443	Stop Admin Config Logs						
	MySQL	7076	3306	Stop Admin Config Logs						

Next, click the Admin button next to MySql (if that fails to do anything enter the following into your browser <u>http://localhost/phpmyadmin/</u>)

Select the **Databases** tab then enter **EmployeeProject** (same name as you specified in QSEE superlite) then click the Create button



Click on the Import tab then Browse and select the sql file you exported from QSEE – then click the Go button

🗊 Databases 📗 SQL 🛛 🚯 Status 🔍 Users 🔂 Exp	oort 📑 Import						
Importing into the current server							
File to Import:							
File may be compressed (gzip, bzip2, zip) or uncompressed. A compressed file's name must end in <b>.[format].[compression]</b> . Example: <b>.sql.zip</b>							
Browse your computer: C:\Users\pjame1sc\MyDc Browse (	Max: 2,048KiB)						
Character set of the file: utf-8							
If all goes well then you will see the following screen:							
🗊 Databases 📗 SQL 🚯 Status 📧 Users 🖬 E	Export 📑 Import 🥜 Settings						
Import has been successfully finished, 8 queries executed. (Er	nployeeProject.sql)						

use EmployeeProject;# MySQL returned an empty result set (i.e. zero rows). -- Create a Database table to represent the "Employee" entity. CREATE TABLE Employee( Emp\_No INTEGER NOT NULL, Emp\_FName VARCHAR(30) NOT NULL, Emp\_SName VARCHAR(30) NOT NULL, Grade TINYINT UNSIGNED NOT NULL,

Select **employeeproject** from the left-hand window:



You will now see the following summary:

🗊 127.0.0.1 » 🗻 employeeproject								
V	Structure	SQL	🔍 Search	间 Query	🔜 Export	🛋 Import	🥜 Opera	
	Table 🔺	Actio	n				Rows	
	employee	BI	rowse 📝 Str	ucture 👒 Sear	ch 👫 Insert 🖗	📄 Empty 🏼 🥥 🛛	Drop	
	emp_on_proje	ect 🔲 Bi	rowse 📝 Str	ucture 👒 Sear	ch 👫 Insert 🦷	📄 Empty 🥥 🛛	Drop	
	pay_structure	BI	rowse 📝 Str	ucture 👒 Sear	ch 👫 Insert 🖗	📄 Empty  🧿 🛛	Drop	
	project	B	rowse 📝 Str	ucture 🍕 Sear	ch 👫 Insert 🖗	📄 Empty 🥥 🕻	Drop	
	4 tables	Sum						

You can examine each of your tables by selecting a table name. For example clicking on Project:

### Project (primary key is underlined)

#	Name	Туре	Collation	Attributes	Null	Default	Extra
1	Proj_Code	varchar(3)	latin1_swedish_ci		No	None	
2	Proj_Desc	varchar(30)	latin1_swedish_ci		No	None	

### Pay\_Structure (primary key is underlined)

#	Name	Туре	Collation	Attributes	Null	Default	Extra
1	Grade	tinyint(3)		UNSIGNED	No	None	
2	Salary	double			No	None	

### Employee (primary key is underlined)

#	Name	Туре	Collation	Attributes	Null	Default	Extra
1	Emp_No	int(11)			No	None	
2	Emp_FName	varchar(30)	latin1_swedish_ci		No	None	
3	Emp_SName	varchar(30)	latin1_swedish_ci		No	None	
4	Grade	tinyint(3)		UNSIGNED	No	None	

# Emp\_On\_Project (primary keys are underlined)

#	Name	Туре	Collation	Attributes	Null	Default	Extra
1	Date_Joined_Project	date			No	None	
2	Months_Allocated_To_Project	tinyint(4)			No	None	
3	Proj Code	varchar(3)	latin1_swedish_ci		No	None	
4	Emp No	int(11)			No	None	

If you look below this table you will see a link called + Indexes . Click on this and you will see any indexes on this table.

- Indexes

Indexes	0									
Action		Keyname	Туре	Unique	Packed	Column	Cardinality	Collation	Null	Comment
🥜 Edit 🏹	Drop	PRIMARY	BTREE	Yes	No	Emp_No	0	Α	No	
🧷 Edit 🐧	Drop	Grade	BTREE	No	No	Grade	0	Α	No	

You can see that Grade has an index (both primary and foreign key fields have indexes).

#### **Click on Relation view**

Print view ♣ Relation view ♣ Propose table structure Track table
and you can see that Grade is a foreign key to Grade in Pay\_Structure.

Relations		
Column	Internal relation 😡	Foreign key constraint (INNODB)
Emp_No		▼.
Emp_FName		No index defined!
Emp_SName		No index defined!
Grade		`employeeproject`.'pay_structure`.'Grade`     ▼     ON DELETE     RESTRICT       ON UPDATE     RESTRICT     ▼